

Improved Genetic Algorithm in SPICE OPUS for Model Parameter Extraction

Arpad Búrmen, Janez Puhan, Tadej Tuma, Iztok Fajfar, Franc Bratkovič

Faculty of Electrical Engineering

University of Ljubljana

Tržaška cesta 25, 1001 Ljubljana, Slovenija

arpadb@fides.fe.uni-lj.si

<http://fides.fe.uni-lj.si/spice>

Abstract

SPICE is a widely used tool for simulation of analog circuits. Accurate models of devices are crucial for obtaining realistic results from the simulation. Although SPICE provides circuit designers with built-in models for most widely used devices many devices still must be modelled with subcircuit models in order to achieve sufficient accuracy. The hardest part of creating a SPICE model for a particular part is determining the parameter values of subcircuit elements. With analytical methods the problem quickly becomes impossible to manage. The only remaining alternative is optimisation. Use of SPICE OPUS optimisation capabilities for model parameter extraction is presented. Three commonly used optimisation methods are examined on examples and discussed in terms of their sensitivity to initial parameter values, speed and numeric stability.

1. Introduction

SPICE [1] is the de-facto standard for analog circuit simulation. As every other CAD tool SPICE too is subject to the known proverb »garbage in-garbage out«. This means that without accurate models it is impossible to obtain realistic simulation results that can be used in the process of circuit design.

Device models are built into the SPICE simulator core for most of the semiconductor devices. Together with model parameters these device models provide the information about behaviour of a particular part or device in the circuit and are often referred to as a "part model".

These built-in device models however do not always provide sufficient accuracy for simulation of many parts widely used in electronic circuits. Lets just take as an example the Zener diode, the power MOS FET or the SCR. There is no built-in device model available for them. The solution to this problem is to model a particular part with a subcircuit that represents the part's behaviour.

Model parameter extraction is used to obtain the values of model parameters from the data provided by the part manufacturer in the datasheet. The procedure is quite straightforward if the equations used to model the part's behaviour are simple. Often determining the most important parameters that characterise a specific part (i.e. I_s , n and R_s for a diode) can be narrowed down to reading a couple of characteristic points from a curve in the datasheet and applying simple formulae to the obtained data. Such models characterise the basic behaviour of a specific part.

However if a more accurate part model is required the above mentioned procedure is not sufficient. In order to model various secondary effects additional devices are added to the "core device" thus producing a subcircuit model. Analytically determining the values of parameters for all elements of such subcircuit is in most cases impossible. The only remaining way is to use various optimisation techniques for curve fitting. Curve fitting also yields more accurate models than various methods based on a couple of characteristic points.

2. State of the Art

Many software packages are available for generating SPICE models. Packages included with commercial SPICE simulators like IsSPICE [2] or PSPICE [3] rely on a few characteristic points from the datasheet and so called »rule of thumb« for determining the parameter values for subcircuit elements. MODPEX [4] from Avant! uses optimisation techniques for curve fitting but there is no support available. Specific software is available for parameter extraction for various models like the BSIM MOS models [5].

Generally whenever a new model for some real-world device or phenomenon is devised an article dealing with model parameter extraction for that specific model is published along with custom tailored software that is capable of extracting parameters from measurement results for that specific model.

In this article we present a unified approach for modelling various real-world devices in SPICE. The approach is based on the SPICE OPUS simulator and its optimisation capabilities. SPICE OPUS is an enhancement of the SPICE3f4 simulator including improved scripting language, optimisation facilities and mixed-mode XSPICE extensions. It was developed by the Group for Computer Aided Circuit Design at Faculty of Electrical Engineering, University of Ljubljana. The Windows95/98/NT, LINUX and SOLARIS release can be downloaded from <http://fides.fe.uni-lj.si/spice>.

3. Methods

The first decision to make is whether to use a numerical analysis package like MATLAB or the built-in SPICE scripting language (NUTMEG) to implement the optimisation procedure. NUTMEG was chosen for the following reasons [6]:

- Using an external package to implement the optimisation procedure greatly increases the

overhead by starting the simulator over and over again whenever a cost function evaluation is required.

- Due to the NUTMEG extensions in SPICE OPUS the loss of flexibility over using external software packages for optimisation is not critical.
- *optimize* command has already been implemented in SPICE OPUS version of NUTMEG

Although the basic SPICE OPUS package includes highly flexible optimisation capabilities some modifications were necessary:

- interpolation of vectors from different plots had to be fixed since it was broken in the original Berkeley release of SPICE
- NUTMEG syntax was extended in order to make importing of vectors from ASCII files possible
- logarithmic compression of search space for a parameter was implemented in the *optimize* command to aid optimisation convergence for parameters which can range over multiple decades
- Since the DC characteristic is one of the most common sources of information in the datasheets DC analysis in SPICE OPUS was improved in order to facilitate logarithmic sweep.

Three optimisation techniques that performed well on test cases [7] were tested:

- constrained simplex algorithm [8]
- Hooke-Jeeves algorithm [9]
- simple genetic algorithm (GA) [10]
- simple GA with elitism

Constrained simplex search and Hooke-Jeeves search both proved to be very efficient optimisation methods in conjunction with SPICE. The constrained simplex and the GA both choose the initial set of points from the parameter space randomly. Hooke-Jeeves algorithm starts from an initial point specified by user.

For the GA Gray encoding was used to encode parameter values in chromosomes in order to moderate the impact of mutation on characteristics of an individual. Population size was 30. Chromosome length was set to 6 bytes per parameter (more than $2 \cdot 10^{14}$ discrete levels per parameter's range). Probability of mutation was set to 0.01 and crossover probability was set to 0.6. Elitism was introduced into the population by automatically copying the best individual into the next generation. The fitness of individuals was based on the cost function in such manner that the fitness of the individual with the lowest cost function value was 5 times greater than the fitness of the worst individual in the population.

The above mentioned optimisation techniques were tested on two curve fitting problems: DC forward characteristic and temperature dependence of a Schottky diode with a guard ring and dynamic resistance characteristic of a Zener diode.

4. Results and Discussion

4.1 Dynamic Resistance of a Zener Diode

Probably the hardest part of building a good Zener diode model is modelling the dynamic resistance vs. reverse current characteristic. The built-in SPICE diode model [11] is virtually useless for this purpose since it models the breakdown with a more or less constant dynamic resistance that is equal to the series resistance of the diode. The dynamic resistance of a Zener diode is important when one wants to obtain realistic results from an AC analysis.

The dynamic resistance can be modelled with two or more parallel branches comprising an ordinary SPICE diode model connected in series to a voltage source. Each branch contributes 4 parameters for fitting (I_s , n and R_s for each diode and V_{DC} for the voltage source). A resistor is connected in series with these parallel branches.

The cost function was defined as the absolute value sum of differences between the logarithm of measured and fitted differential resistance.

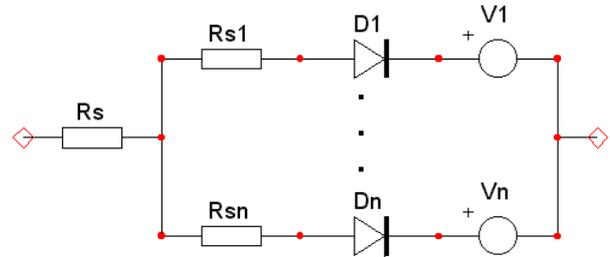


Figure 1 : Zener diode reverse characteristic model.

For the purpose of comparing various optimisation algorithms The Zener diode was BZX55 9V1. Reverse characteristic was modelled with two parallel branches. Two sets of initial values were tested. Initial values of parameters were chosen as follows:

$$\text{Initial 1: } (RS, RS_1, RS_2, IS_1, IS_2, n_1, n_2, Vdc_1, Vdc_2) = (1, 1, 1, 10^{-12}, 10^{-12}, 1, 1, 0.9, 1.1)$$

$$\text{Initial 2: } (RS, RS_1, RS_2, IS_1, IS_2, n_1, n_2, Vdc_1, Vdc_2) = (1, 1, 1, 10^{-9}, 10^{-9}, 1, 1, 0.9, 1.1)$$

	Initial 1	Initial 2
Number of iterations	516	427
Optimisation time	17.6s	15.0s
Initial cost function value	3.65758	3.65757
Minimal cost function	0.353	0.458

Table 1: comparison of Hooke-Jeeves algorithm performance for two initial points.

As it can be seen from Table 1 the Hooke-Jeeves algorithm's performance in terms of the minimal cost function depends on the initial point due to local nature of the method.

Table 2 is the summary of performance for the constrained simplex method for the same search space as for the Hooke-Jeeves method. By default the number of points in the simplex (8) is twice the number of parameters that are to be optimised. The points are

selected randomly from the search space so no initial point needs to be provided explicitly.

The constrained simplex method performed worse than Hooke-Jeeves. The minimal cost function didn't improve even if the number of points in the simplex was increased.

Points in simplex	18	30	60
Cost function evaluations	973	2077	4257
Optimisation time	30.5s	67.2s	130.7s
Minimal cost function	0.769	0.768	0.769

Table 2: performance of the constrained simplex algorithm.

Table 3 summarises the performance of the simple GA without elitism. As it can be seen the results are not very encouraging although they are a bit better than the ones obtained with the constrained simplex method.

Population size	10	20	30
Cost function evaluations	1210	2420	3630
Optimisation time	40.5s	76.0s	112.5s
Minimal cost function	0.697	0.778	0.635

Table 3: performance of the GA without elitism after 120 generations. $p_{cross}=0.6$, $p_{mutation}=0.01$.

Figure 2 explains the poor performance of the GA. Around generation 200 better individuals start to emerge. However due to mutation and crossover they vanish around generation 240. One solution to this problem would be to decrease the probabilities of crossover and mutation. This could slow down the evolution, which is slow enough already.

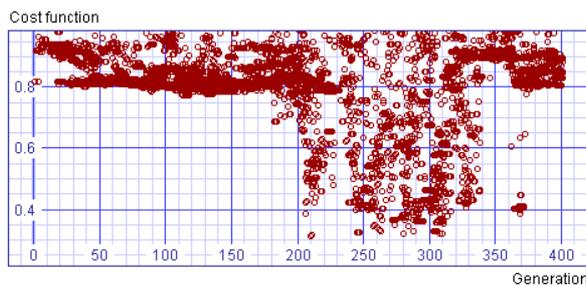


Figure 2: the population evolved by the simple GA without elitism (population size 20).

In order to improve convergence, elitism was introduced into the simple GA. The results for the modified GA are in Table 4. The best individual after 120 generations is even better than the one obtained by the Hooke-Jeeves search.

There are vast differences between best individuals for various population sizes. Best results were obtained with a population a bit larger than twice the number of parameters.

Population size	10	20	30
Optimisation time	39.4s	75.6s	111.9s
Minimal cost function	0.667	0.211	0.597

Table 4: performance of the GA with elitism after 120 generations. $p_{cross}=0.6$, $p_{mutation}=0.01$.

Figure 3 shows the evolution of the population with the modified GA. As it can be seen the best individual emerges somewhere around generation 100. Elitism helps to preserve the best individual from disappearing with crossover. The result of the optimisation is shown in

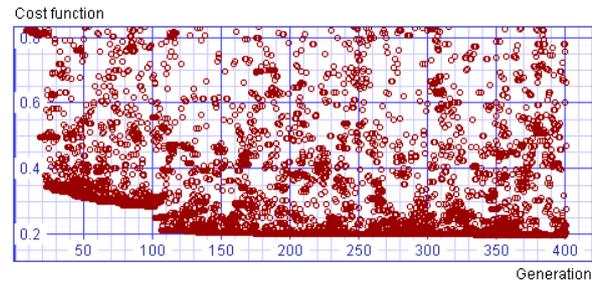


Figure 3: the population evolved by the simple GA with elitism (population size 20).

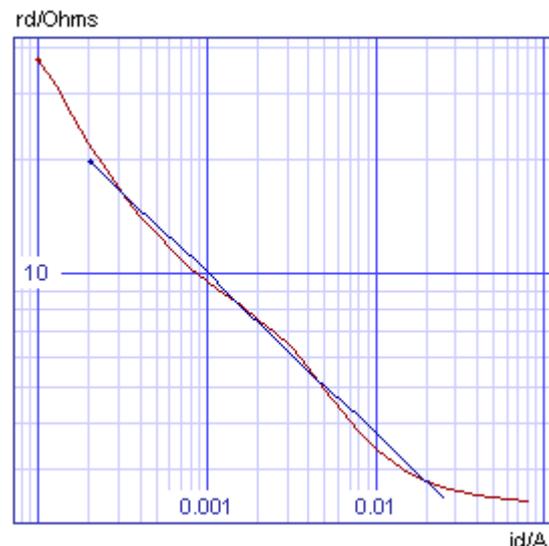


Figure 4: result of parameter extraction with the modified GA. The shorter curve represents values from the datasheet.

4.2 Shottky Diode With a Guard Ring

A Shottky diode with a guard ring has a bit different characteristics than an ordinary semiconductor diode. The built-in SPICE diode model can't be fitted very well to the DC characteristic of such diode. The solution is to model the diode with a temperature dependent resistor and two SPICE diodes connected in parallel (Figure 5). This doubles the number of parameters (12: I_s , n , E_g , T_{nom} and XTI for each diode and R_s and TCl for the resistor). Due to the nonlinear nature of the problem it is impractical to determine the parameters analytically. The parameters are fitted to two I_D/V_D DC characteristics at different temperatures.

Cost function was calculated for each curve as the absolute value sum differences between the logarithm of

the expected and the fitted current. Then the largest of these sums was taken as the cost function.

The diode used was SB020 from General Semiconductor. Since it is impractical to try multiple initial points in order to find a good fit, Hooke-Jeeves search was not applied to this problem. Only constrained simplex method and modified GA were tested.

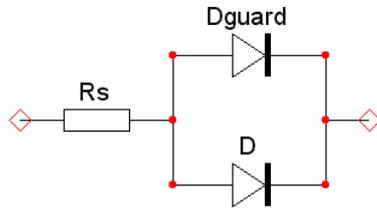


Figure 5: model used for fitting the DC characteristic of a guard-ring Shottky diode.

Constrained simplex required 1003 cost function evaluations (40.2s) to find the optimal fit (cost function value 0.622).

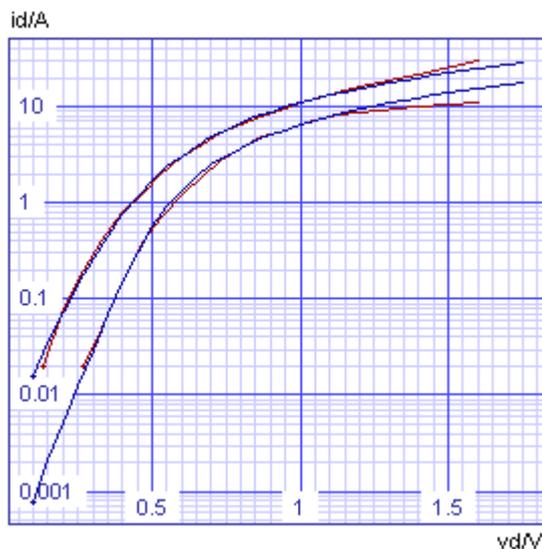


Figure 6: result of parameter extraction with the modified GA. Shorter curves stand for the values from the datasheet.

For the modified GA the same encoding and probabilities for crossover and mutation were used as in the previous example. Population size was set to 30. An individual with cost function value 0.743 emerges after 80 generations. After 120 generations (3630 cost function evaluations, 143.5s) the best individual's cost function value becomes 0.662 which is not far from the solution obtained by the constrained simplex method. Figure 6 shows the result obtained by the modified GA.

5. Conclusion

The use optimisation algorithms in SPICE OPUS for model parameter extraction was demonstrated on two examples. The results show that even a robust

optimisation algorithm such as constrained simplex can yield unsatisfactory results.

The use of a simple GA was presented. Due to its bad performance elitism was added to the GA in SPICE OPUS. The modified algorithm showed significant improvement in convergence speed over the original one and is capable of delivering results that are comparable or better than the ones obtained by the constrained simplex method.

6. References

- [1] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, "SPICE3 Version 3f4 User's Manual", University of California, Berkeley, California, 1989
- [2] <http://www.intusoft.com/spicemod.htm>
- [3] http://www.pspice.com/product_info/pspicedsr9.asp#editor
- [4] <http://www.symmetry.com/modpex.html>
- [5] J. R. Pierret, "A MOS Parameter Extraction Program for the BSIM Model", Rep. No. ERL-M84/99, Electronics Research Laboratory, University of California, Berkeley, 1984
- [6] J. Puhon, T. Tuma, "Optimization of analog circuits with SPICE 3f4", Proceedings of the ECCTD'97, Volume 1, pages 177 - 180, 1997
- [7] J. Puhon, T. Tuma, I. Fajfar, "Optimisation Methods in SPICE, a Comparison", Proceedings of the ECCTD'99, Volume 1, pages xxx-xxx, 1999
- [8] M. J. Box, "A new method of constrained optimisation and comparison with other methods", Computer Journal, Volume 7, pages 42 - 52, 1965
- [9] R. Hooke, T. A. Jeeves, "Direct search" solution of numerical and statistical problems", Journal of ACM, Volume 8, pages 212 - 229, 1961
- [10] David E. Goldberg, "Genetic Algorithms in Search, Optimisation & Machine Learning", Addison-Wesley Publishing, 1989
- [11] G. Massobrio, P. Antognetti, "Semiconductor Device Modeling with SPICE", McGraw-Hill, 1993

